

TP 5 : Recalage d'images

Dans ce TP nous allons nous familiariser avec les différents outils fournis dans la toolbox Matlab *Image Processing* pour le recalage d'images. Nous avons vu en cours qu'il existait de très nombreuses transformations (linéaires, non linéaires, qui conservent certaines propriétés de l'image, etc). Dans un premier temps, nous nous intéresserons en particulier à la rotation et aux différentes techniques d'interpolation que l'on peut utiliser. Dans un second temps, on utilisera les outils permettant d'estimer une transformation pour recaler deux images d'une même scène et on mesurera la qualité du recalage.

1 Rappels théoriques et instructions Matlab

Parmi toutes les transformations évoquées en cours, nous allons nous intéresser aux transformations affines. On présente dans cette partie quelques rappels théoriques sur la modélisation des transformations affines et les fonctions Matlab que nous utiliserons dans le TP.

1.1 Transformations affines

Parmi les transformations affines, on trouve :

- les translations
- les changements d'échelles
- les rotations
- les cisaillements
- une combinaison de ces transformations.

La transformation affine s'exprime comme une transformation des coordonnées $\vec{p} = [x, y]$ des pixels de l'image de référence pour donner les coordonnées $\vec{p}' = [x', y']$ de ces pixels dans l'image déformée. Cette transformation peut s'écrire :

$$\vec{p}' = \vec{p}M + \vec{t} \quad (1)$$

où M est une matrice de dimension 2×2 codant une transformation linéaire et $\vec{t} = [u, v]$ est la translation entre le point invariant des deux images au sens de la transformation M . A noter que dans le cas d'une translation simple, la matrice M est égale à la matrice identité.

La transformation peut aussi être exprimée comme une seule multiplication matricielle :

$$[x', y', 1] = [x, y, 1] \begin{pmatrix} M_{1,1} & M_{1,2} & 0 \\ M_{2,1} & M_{2,2} & 0 \\ u & v & 1 \end{pmatrix} \quad (2)$$

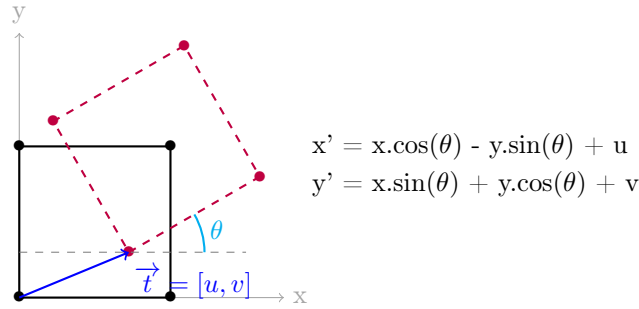


FIGURE 1 – Exemple de transformation affine : translation de vecteur \vec{t} + rotation d'angle θ

1.2 Commandes Matlab pour effectuer des transformations affines

Voici quelques commandes Matlab utiles pour effectuer des translations, des rotations et des changements d'échelles de l'image :

```
>> B = imtranslate(A,translation);
>> B = imrotate(A,angle, method, bbox);
>> B = imresize(A,scale);
```

Pour appliquer une transformation quelconque à l'image A, il faut utiliser la commande suivante :

```
>> B = imtransform(A,tform);
```

Il faut que **tform** soit une structure Matlab de type **TFORM** pour cela, on peut utiliser la fonction Matlab **maketform** pour créer une structure dans laquelle on peut renseigner la matrice de transformation

$$F = \begin{pmatrix} M_{1,1} & M_{1,2} & 0 \\ M_{2,1} & M_{2,2} & 0 \\ u & v & 1 \end{pmatrix}.$$

```
>> tform = maketform('affine',F);
```

On peut retrouver cette matrice F de transformation en écrivant :

```
>> F = tform.tdata.T
```

Par exemple pour un cisaillement, on définit la matrice :

$$F = \begin{pmatrix} 1 & \beta & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

et on transforme cette matrice en structure **TFORM** avec la fonction **maketform**.

On verra dans la section 3.2 qu'on peut aussi estimer directement une transformation géométrique grâce aux fonctions Matlab.

2 Transformation et interpolation

Le but de cette section est de mettre en évidence de l'interpolation effectuée lors de certaines transformations. Nous allons étudier en particulier la rotation qui, en théorie, conserve les angles, le parallélisme et les distances, en effet, dans le cas continu (l'image est considérée comme une fonction continue définie sur \mathbb{R}^2), la rotation ne modifie pas le contenu de l'image. Nous allons étudier ici l'effet de la rotation dans le cas d'une image numérique. La rotation s'effectue sous Matlab avec la fonction **imrotate**.

Manipulation 1

- ➡ Charger l'image 'briques.jpg'.
- ➡ Transformer l'image en niveaux de gris avec la commande :

```
>> I = rgb2gray(imread('briques.jpg'));
```

- ➡ Effectuer 4 rotations successives de 90° de cette image (on pourra observer l'effet de la rotation à chaque étape). On utilisera les options 'nearest' pour l'argument `method` et 'crop' pour l'argument `bbox` dans la fonction `imrotate`.
 - ➡ Comparer l'image obtenue avec l'image originale. On peut visualiser la différence et calculer l'EQM sur la zone centrale de l'image.
 - ➡ Effectuer maintenant 6 rotations de 60° et comparer avec l'image originale. Qu'observe-t-on ?
 - ➡ On peut également tester d'autres méthodes d'interpolation que 'nearest' pour l'argument `method` dans la fonction `imrotate` : 'bilinear' et 'bicubic' et recalculer l'EQM dans le cas des 6 rotations de 60° .
-

3 Réalisation d'une mosaïque d'images

On dispose ici de deux observations par avion de la ville de Toulouse produite par le CNES, ces deux observations se recouvrent partiellement. On souhaiterait recaler ces deux images pour construire une vue plus large de la ville. Les attributs utilisés pour le recalage sont des points de contrôle sélectionnés manuellement par l'utilisateur. Le but de cette partie est de prendre en main les outils Matlab pour le recalage et de comprendre l'influence du choix des points de contrôle sur la qualité de la reconstruction.

3.1 Recalage par point de contrôle : sélection des points

La toolbox *Image processing* de Matlab fournit une interface graphique permettant de sélectionner les points de contrôles manuellement sur l'image à recaler `im2` et sur l'image de référence `im1`. L'interface graphique se lance grâce à la commande suivante :

```
>> cpselect(im2, im1);
```

Une fois les points sélectionnés, il faut les stocker dans une structure appropriée pour cela, il faut sélectionner : `File` → `ExportPointstoWorkspace`. Attention à l'ordre de sélection des points entre les deux images ! Par défaut, les coordonnées des points sur les deux images sont stockées dans les variables `input_points` et `base_points` qui seront ensuite utilisées pour estimer la transformation.

Manipulation 2

- ➡ Utiliser la commande `cpselect` pour sélectionner 3 points de contrôle sur les images : `im1 = CNES_Toulouse_ref.jpg` et `im2 = CNES_Toulouse2.jpg`.
 - ➡ Les variables de stockage `input_points` et `base_points` seront conservées pour la suite du TP.
-

3.2 Estimation de la transformation géométrique

L'estimation de la transformation se fait grâce à la fonction `cp2tform`. Aller voir dans l'aide les différentes options à passer par paramètres à cette fonction. Le code suivant permettra d'estimer la transformation et recaler les deux images `im1` et `im2` dans la suite du TP dans le cas d'une transformation affine :

```
>> t_affine = cp2tform(input_points, base_points, 'affine');
>> [M1, N1] = size(im1);
>> [M2, N2] = size(im2);
>> recalage = imtransform(im2, t_affine, 'XData', [1, M1+M2], 'YData', [1,
N1+N2]);
>> imshowpair(im1, recalage);
```

La fonction `cp2tform` permet d'estimer la transformation à appliquer à l'image déformée `im2` pour la recaler sur `im1` à l'aide des points de contrôle sélectionnés, la matrice de transformation est stockée dans la variable `t_affine.tdata.T`.

Manipulation 3

- ➡ Estimer la transformation géométrique à partir des 3 points de contrôles sélectionnés précédemment. On considérera que la transformation est affine.
 - ➡ Observer la forme de la transformation et le résultat du recalage.
-

3.3 Qualité de l'estimation de la transformation

A partir de la manipulation précédente, on veut estimer la qualité du recalage. Pour cela on définit l'image de la différence :

```
>> difference = im1 - recalage;
```

Afin d'estimer la qualité de la transformation, on va calculer l'erreur quadratique moyenne du recalage. Pour cela, il faut sélectionner une zone commune des deux images `im1` et `recalage` par exemple la zone `[220:380, 220:380]`.

Manipulation 4

- ➡ Calculer l'erreur quadratique moyenne sur la zone `[220:380, 220:380]` pour le recalage effectué avec les 3 points de contrôle précédents.
 - ➡ Recommencer l'étape de recalage avec 5 points de recalage et calculer l'EQM obtenue.
-

On peut aussi étudier l'influence de la position des points de contrôle dans l'image.

Manipulation 5

- ➡ Effectuer un recalage avec 5 points de contrôle localisés sur une zone restreinte de l'image (par exemple au niveau de la place du Capitole). Calculer l'EQM.
 - ➡ Recommencer le recalage avec 5 points de contrôle étalés sur toute l'image. Calculer l'EQM et comparer avec la valeur obtenue précédemment.
 - ➡ Observer visuellement la qualité du recalage dans les deux cas.
-

4 Pour aller plus loin ... (facultatif)

Manipulation 6

- ➡ A partir de la meilleure estimation de la transformation obtenue, déduire l'angle de rotation subie par l'image `CNES_Toulouse2.jpg`
-